[19]中华人民共和国国家知识产权局

[51]Int. Cl[7]
G06F 1/00

[12]发明专利申请公开说明书

[21]申请号 99805763.0

[43]公开日 2001年6月13日

[11]公开号 CN 1299478A

[22]申请日 1999.2.25 [21]申请号 99805763.0
[30]优先权
[32]1998.3.2 [33]IL [31]123512
[86]国际申请 PCT/IL99/00113 1999.2.25
[87]国际公布 WO99/45454 英 1999.9.10
[85]进入国家阶段日期 2000.11.2
[71]申请人 电脑相关想象公司
地址 美国纽约
[72]发明人 D·埃尔格雷斯 F·本阿德雷特

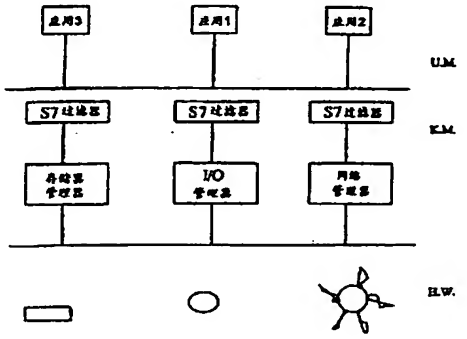
[74]专利代理机构 中国专利代理(香港)有限公司
代理人 吴立明 王忠忠

权利要求书2页 说明书5页 附图页数3页

[54]发明名称 用于保护以防未授权使用计算机资源的方法和代理

[57]摘要

防止工作站内运行的应用程序对计算机资源恶意使用的方法和代理。确定未指定的应用程序不允许访问的服务的列表，而且当这种未指定的应用程序在工作站内运行时，应用程序被阻止直接访问任何资源。对特定服务的访问的任何直接或间接的请求被分析，以确定根据此服务列表是否这种请求是允许的。如果它是允许的，工作站就处理该请求。如果请求是不允许的，则阻止未指定的应用程序访问被请求的资源。资源可以是任何本地资源或远程资源，例如存储器分配，文件，目录，对目录和文件的操作如复制、删除或压缩，或其它导致工作站或其外围设备永久改变的操作。查询表包括未指定的应用程序不允许访问的服务列表，该表用于确定未指定的应用程序生成的直接或间接的请求是否是允许的。代理包括预先设定的应用程序列表，该列表包括各种应用程序可以利用的资源列表。



ISSN1008-4274

知识产权出版社出版

权 利 要 求 书

1. 一种防止工作站内运行的应用程序程序对计算机资源的恶意使用的方法，该方法包括下列步骤:

a）提供服务列表，未指定的应用程序不允许访问这些服务;

b）当这种未指定的应用程序在工作站内运行时，防止该应用程序直接访问任何资源;

c）分析访问特定服务的任何直接或间接的请求，根据上面 a）中定义的服务列表确定这种请求是否是被允许的;

d）如果请求是允许的，就允许工作站处理它；和

e）如果请求是不允许的，就禁止未指定的应用程序访问被请求的资源。

其中该资源可能是任何的本地或远程资源，包括，但不限于，存储器分配，文件，目录，对文件和目录的操作如复制、删除或压缩，或任何其它的会导致工作站或其外围设备中发生改变的操作。

2. 根据权利要求 1 中的方法，其中服务列表作为查询表而被提供。

3. 根据权利要求 1 或 2 中的方法，其中未经指定的应用程序是在预先设定的应用程序列表中未被特别定义的应用程序。

4. 根据权利要求 3 的方法，其中预先设定的应用程序列表包括各种应用程序都可利用的资源列表。

5. 一种保护工作站以防该工作站内运行的未经指定的应用程序对计算机资源恶意使用的代理，包括:

a）检测在工作站中运行的未经指定的应用程序的装置;

b）确定被该未经指定的应用程序使用的资源的请求的装置;

c）识别对资源利用的链式请求的装置，其中该链式请求包括被未经指定的应用程序所调用的资源生成的请求;

d）确定该未经指定的应用程序直接生成的请求是否是允许的装置;

e）确定该未经指定的应用程序间接生成的作为链式请求的请求是否是不允许的装置;

f）如果确定请求是不允许的或是未经指定的应用程序所直接生

成的请求是不允许的，那么用于防止该链式请求被处理，否则就允许其被处理的装置。

6. 根据权利要求 5 的代理，其中用于确定该未经指定的应用程序直接或间接生成的请求是否是允许的装置包括一个查询表，该表包括未经指定的应用程序不允许访问的服务的列表。

7. 根据权利要求 5 或 6 的代理，其中该资源可以是，但并不限于，任何的本地或远程资源，包括存储器分配，文件，目录，对文件和目录的操作如复制、删除或压缩，或任何其它会导致工作站或其外围设备中永久改变的操作。

8. 根据权利要求 5 至 7 中任一个的代理，包括应用程序预先设定的列表，它包含每个应用程序可利用资源的列表。

00.11.02

说　明　书

用于保护以防未授权使用计算机资源的方法和代理

发明领域

本发明涉及计算机安全管理。具体涉及一种用于防止通过恶意的应用程序对计算机资源使用的访问的方法和代理。

发明背景

自从几年前开始，因特网在内容和应用程序技术方面都发展很大。在因特网初期，网站仅包括文本，不久以后就引入了图像。随着因特网的发展，许多压缩标准，如图像、声音和录像文件，连同用于播放它们的程序（称为播放器）都得到了发展。最初，这种文件在用户的请求下被下载到用户工作站，并只能由由适当的播放器在用户的特定的要求下解压。

在环球网发展的自然进程中，当开始寻找一种方法能显示出更好的、交互式的、生动的网页时，Sun Microsystems Inc 开发了 Java，一种语言允许万维网站点管理员编写程序（一列命令－网络可执行命令），多数时候被下载到用户工作站上而该用户并不知道，并被其工作站上的浏览器执行。这些可执行命令用于，如提供图像动画和网络漫游者屏幕上的其它图像。这种可执行命令含有接近用户工作站资源的途径，这将导致重大的安全问题。尽管某种程度的安全在 Java 语言中得到定义，但是不久巨大的安全漏洞就在此语言中被发现。

自从开发了 Java 后，Microsoft 开发了 ActiveX，它是另一种网络可执行格式，也被下载到工作站上。ActiveX 也具有同类的安全问题。

因特网上充满了网络可执行命令，用户在知道或不知道的情况下被下载到组织内的工作站上。这些代码通常含有无害的函数。尽管通常是安全的，但是他们不能满足组织内被要求的安全策略。

一旦执行，代码可能堵塞网络，引起对本地数据库，工作站和服务器的严重的不可恢复的破坏，或导致对来自服务器/工作站的信

息未经授权的检索。这些部件可能出现在 Java 小应用程序程序、ActiveX 组件、动态连接库和其它目标代码中，它们的使用正在以不平行的速度增加。这些小应用程序程序中的大多数未经请求和未经控制被下载到组织中。企业没有办法了解它们的存在或执行，而且没有系统适用于早期检测并防止代码被执行。

在某些情况下，由于内联网和局域网的存在，该问题被恶化，这种网络可以被未经授权的人员使用访问工作站并在其中实行恶意的行为。

安全问题被浏览器制造商部分地解决了，制造商让用户禁止使用可执行程序。当然这不是合理的解决方法，因为所有的电子商务和广告都是基于可执行命令的使用。

在同一申请人的三个共同未决的专利申请中，1997 年 3 月 10 日提交的 IL 120420，1997 年 9 月 22 日提交的 IL 121815，和 1997 年 11 月 27 日提交的 IL 122314，它们整体在此作为参考。描述了用于防止不需要的可执行目标渗透到我们工作的局域网/广域网和最终的工作站和服务器的方法和装置。IL 122314 更进一步地提供了一种方法，用于加强安全策略，有选择性地防止在个人工作站中下载和执行非期望的可执行目标。

尽管在上述提到的保护个人工作站的专利申请中已经做了许多工作，但是仍有一个问题待于解决：通过已经通过任何初期安全检查（如网关安全策略）的应用程序而对本地资源恶意地使用。因为这些应用程序未违反这种安全策略，或是未通过初期的检查点（如前面提到的以色列的专利申请中说明的配有安全策略检查的网关）的应用程序，因为这种初期检验点是不可用的，或是因为这些应用程序被直接在工作站上下载。对 CPU 资源这种恶意的使用会导致对工作站的数据、操作和硬件的破坏，在上述考虑到的情况下，直到破坏完成才能被发觉。

本发明的一个目的是提供一种能够克服现有技术方法前述的缺点在工作站一级上提供有效的保护的方法和代理。

本发明的另一个目标是提供一种能够被用于有效地防止通过工作站内运行的应用程序而对该工作站资源恶意的使用的方法和代理。

该发明的其它目标和优势在下面的说明中显而易见。

发明概述

一方面，本发明是一种用于防止工作站内运行的应用程序对计算机资源恶意使用的方法。该方法包括以下步骤：

a）提供服务列表，未指定的应用程序不允许访问这些服务；

b）当这种未指定的应用程序在工作站内运行时，防止该应用程序直接访问任何资源；

c）分析访问特定服务的任何直接或间接的请求，根据上面 a）中定义的服务列表确定这种请求是否是被允许的；

d）如果请求是允许的，就允许工作站处理它；和

e）如果请求是不允许的，就禁止未指定的应用程序访问被请求的资源。

其中该资源可能是任何的本地或远程资源，包括，但不限于，存储器分配，文件，目录，对文件和目录的操作如复制、删除或压缩，或任何其它的会导致工作站或其外围设备中发生改变的操作。说明的但未限定的这种操作的实例包括访问系统文件、配置信息、网络通信、硬件设备（磁盘、调整解调器等）、CMOS 数据（时间、日期等），或是使用资源如存储器分配，过程创建，线程创建，以及使用剩余的 CPU 时间，剩余的磁盘空间，剩余的网络通信，剩余图形资源和系统或应用程序配置。

根据本发明的优选实施方案服务列表作为查询表被提供。

“未指定的应用程序”是指未被特别定义在预先设定的应用程序列表中的应用程序。根据发明的优选实施方案，该预先设定的应用程序列表包括每个应用程序可利用资源的列表。

另一方面，本发明是一种用于保护工作站以防该工作站内运行的未指定的应用程序对计算机资源恶意使用的代理。该代理包括：

a）用于检测在工作站中运行的未经指定的应用程序或应用程序模块的装置；

b）用于确定被该未指定的应用程序使用的资源的请求的装置；

c）用于识别对资源利用的链式请求的装置，其中该链式请求包括被未经指定的应用程序所调用的资源生成的请求；

d）用于确定该未经指定的应用程序直接生成的请求是否是允许

的装置；

e）用于确定该未经指定的应用程序间接生成的作为链式请求的请求是否是不允许的装置；

f）如果确定请求是不允许的或经未指定的应用程序所直接生成的请求是不允许的，那么用于防止该链式请求被处理，否则就允许其被处理的装置。

根据本发明的优选实施方案，用于确定是该未经指定的应用程序直接或间接生成的请求是允许的装置包括查询表，表中包括未经指定的应用程序所不允许访问的服务的列表。在发明的另一优选实施方案中，代理包括预先设定的应用程序列表，该应用程序列表包括可以被各种应用程序利用的资源的列表。

上述提到的和许多其它的本发明的特征和优点可以参考附图通过下面优选实施方案的说明性而非限定性的实例得到更好的理解。

附图简述

图 1 示意地说明了不同的应用程序和它们的请求及相关的操作；

图 2 示意地说明该能导致机器出错的应用程序的细节；

图 3 说明尝试间接的不允许的资源利用的情况。

优选实施方案详细说明

这些情况的实例在图 1—3 中被示出。参考图 1，三个不同的应用程序被表示出来，被标为“应用 1”到“应用 3”。进程发生在三个不同的层次上：用户模式（标识为“U.M.”），内核模式（标识为“K.M.”）和硬件（标识为“H.W.”）。这三个不同的模式在图上用直线示意性的分离开。“应用 1”、“应用 2”和“应用 3”应用程序在用户模式下操作。“应用 1”是“打开文件”输入/输出请求。这一请求被传递到输入/输出管理器，此管理器反过来引用磁盘执行这一被请求的操作。过滤器（在图中标识为“S7 过滤器”）分析请求以确定根据安全策略它是否允许的。如果是允许的，它就被允许继续进行到输入/输出管理器，它处理对磁盘的请求。

另一方面，“应用 2”生成一个包括网络的请求，即“打开到文件服务器的连接”的请求。仅仅在过滤器 S7 确定请求被允许时，网络管理器才被允许处理这一请求。同样地，“应用 3”生成存储器分配请求，该请求被过滤器检测，如果是被允许的，它就被传递到存

储器管理器然后对存储器进行操作。

内核模式和与此相对的硬盘中各种请求的操作，在经过过滤器检测并允许它们以后，与日常计算机的传统操作是一致的，并为本领域的技术人员所熟知，因此为了简化这里不再详细说明。

参看图 2，能引起机器出错的示例性应用程序的细节被示出。在这一实例中，“应用 1”生成 1000 个新的进程的请求。如果本发明中的系统不存在，这 1000 个请求将被过程管理器传递给 CPU，并使用 CPU 的所有资源，这样阻塞机器的操作。然而，如果发明中的过滤器存在，过滤器可以预先设定同一应用程序允许生成仅仅有限数量的进程。因此，如果大量新的进程被单一的应用程序所请求，这些进程超出了预先设定的界限，那么过滤器 S7 将不允许该应用程序向过程管理器传递，这样以避免耗尽机器资源。

图 3 说明了尝试间接的不被允许的资源利用的情况。在这一实例中，“应用 1”是一种不允许把请求发送给输入/输出管理器的类型。如果“应用 1”尝试这样做，它将被 S7 过滤器阻断，除非请求遵循 S7 预先设定的安全策略。因此，“应用 1”可以被编程以便实现过程间通信，即将其请求通知给更进一步的进程 APPX，APPX 被允许生成“应用 1”所不允许生成的对输入/输出管理器的请求。在这种情况下，用户模式和内核模式之间的 S7 过滤器被绕过。为了防止这种情况，更进一步的过滤器 S7 被设置在所有的通信进程之间，并阻止被从一个过程传递到另一个过程的任何请求（在例子中，从“应用 1”到 APPX），这种请求在第一个过程中不被允许直接生成。

当然，过滤器 S7 不是物理过滤器，而是逻辑过滤器，这一点对于技术人员是显然的。通过使用多种不同的、根据系统涉及的特定要求由技术人员预先确定的分析过程和标准，这种逻辑过滤器可以以多种方式实现。

因此，所有上面的说明和实例仅仅为了说明而被提供，除附加的权利要求所定义的以外，并非意欲在任何方面限定本发明。

说 明 书 附 图

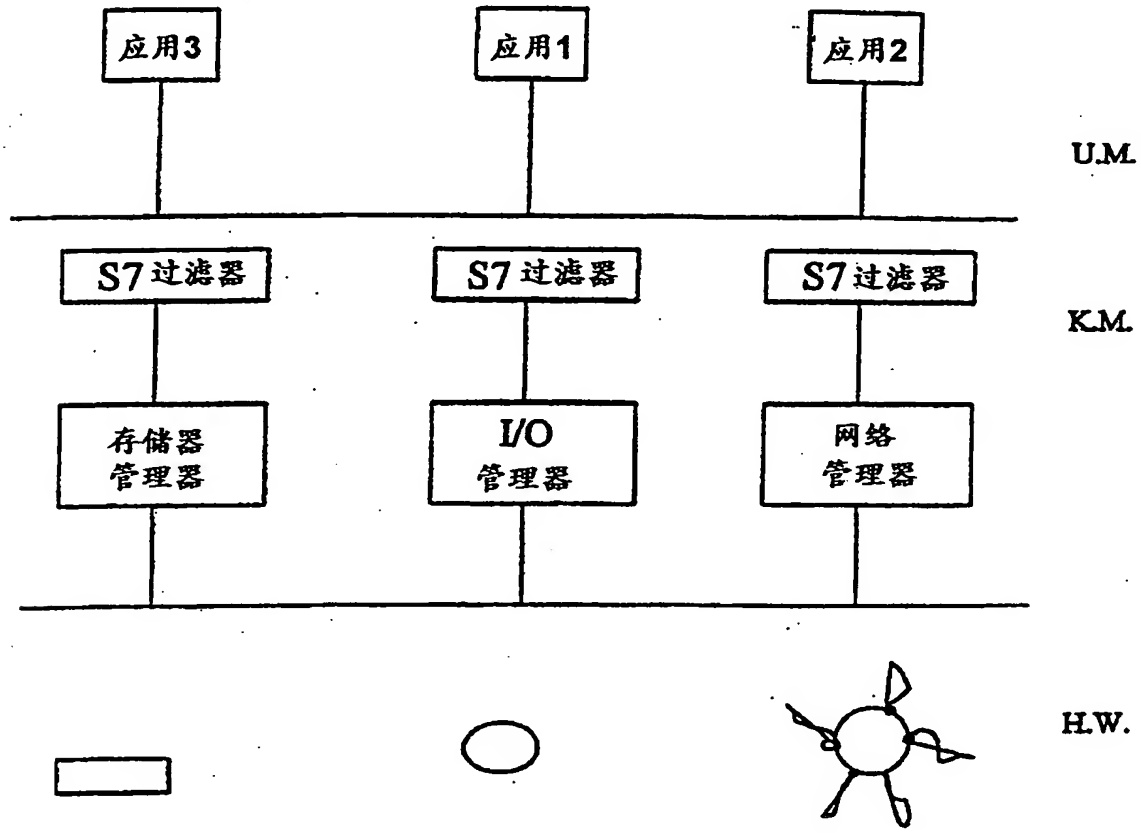


图 1

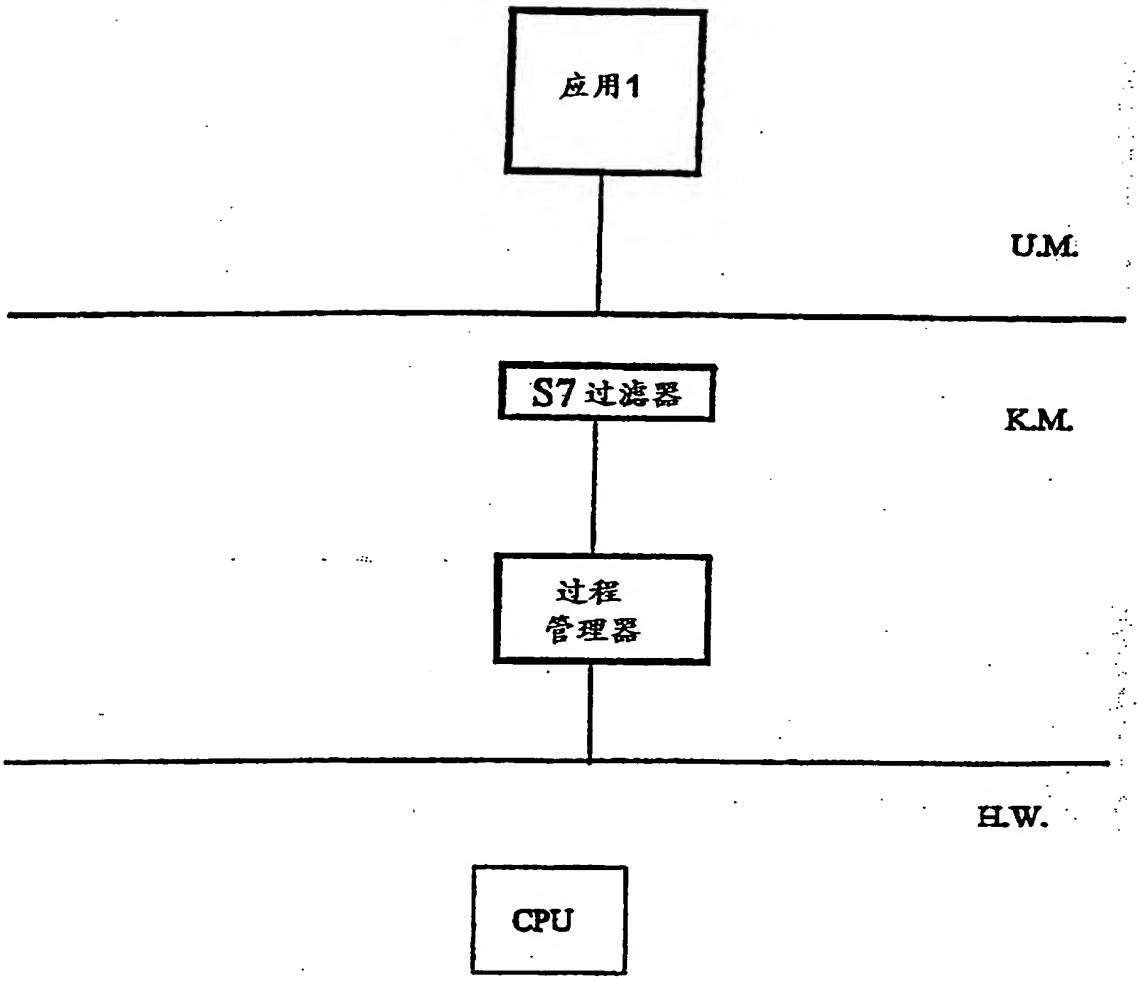


图 2

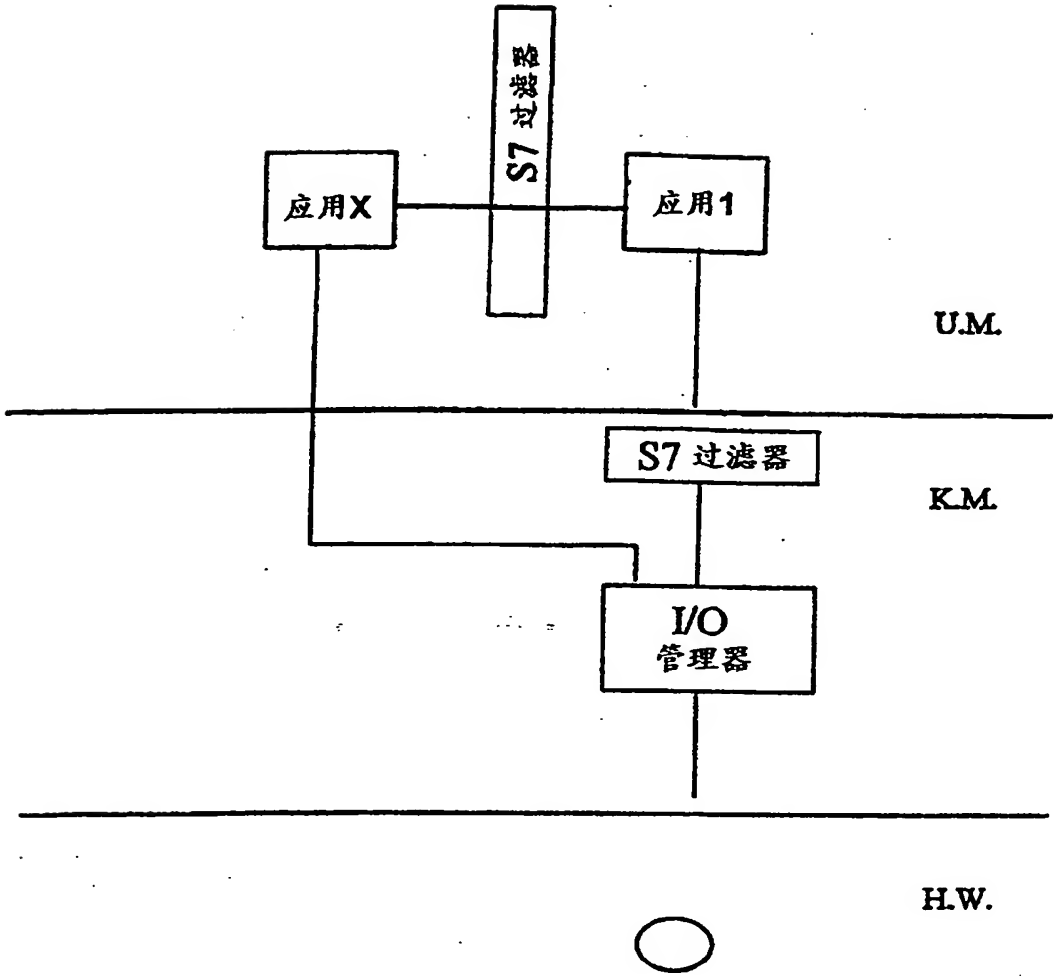


图 3